

COMPLETE SET OF PENDING CLAIMS

1 1-29. (Cancelled)

1 30. (Previously Presented) A data storage method that stores a virtual
2 machine instruction sequence generated by compiler to be executed by a virtual machine,
3 the data storage method comprising:

4 dividing the virtual machine instruction sequence into basic blocks each
5 corresponding to an instruction block;

6 transmitting the instruction block to the virtual machine;

7 storing the instruction block in the virtual machine; and

8 formatting the instruction block to include:

9 an identifier area for storing an identifier that specifies a start position of
10 the instruction block;

11 a non-branch instruction area for storing non-branch instructions
12 belonging to the corresponding basic block;

13 a branch instruction area for storing at least one branch instruction
14 belonging to the corresponding basic block; and

15 each branch instruction stored in the branch instruction area designating a
16 branch destination using an identifier stored in one of the identifier areas.

1 31-38. (Cancelled)

1 39. (Previously Presented) A data storage method that stores a virtual
2 machine instruction sequence generated by compiler to be executed by a virtual machine,
3 the data storage method comprising:

4 dividing the virtual machine instruction sequence into basic blocks each
5 corresponding to an instruction block;
6 transmitting the instruction block to the virtual machine;
7 storing the instruction block in the virtual machine;
8 formatting the instruction block to include:
9 an identifier area for storing an identifier that specifies a start position of
10 the instruction block;
11 a non-branch instruction area for storing non-branch instructions
12 belonging to the corresponding basic block; and
13 a branch instruction area for storing at least one branch instructions
14 belonging to the corresponding basic block.

1 40. (Previously Presented) A data storage method of Claim 39,
2 wherein the identifier of the instruction block is an address related information in
3 the virtual machine instruction sequence.

1 41. (Previously Presented) A data storage method of Claim 40,
2 wherein the address related information is one of absolute address, relative
3 address, and offset address.

1 42. (Previously Presented) A data storage method of Claim 40,
2 wherein whether each virtual machine instruction is positioned at a start position
3 of the basic block is indicated by an address in the virtual machine instruction sequence
4 to which the virtual machine instruction is allocated;
5 a virtual machine instruction at the start position of the basic block being
6 allocated to a specific address in the virtual machine instruction sequence, and

7 a virtual machine instruction at other than the start position of the basic block
8 being allocated to an address other than the specific address.

1 43. (Previously Presented) A data storage method of Claim 40,
2 wherein virtual machine instructions of the virtual machine instruction sequence
3 each comprises:
4 an identifying unit for storing identification information which indicates if the
5 virtual machine instruction is positioned at a start position of the basic block; and
6 an operation specifying unit for specifying an operation to be executed by the
7 virtual machine.

1 44. (Previously Presented) A data storage method of Claim 40,
2 the basic blocks; and
3 identification tags, each designates an address related information of the virtual
4 machine instruction at a start position of the basic block; attachment of the tag indicating
5 if the virtual machine instruction corresponding to the identification tag is positioned at
6 the start position of the basic block.

1 45-51. (Cancelled)

1 52. (Previously Presented) A method of storing a virtual machine
2 instruction sequence generated by compiler to be executed by a virtual machine, the
3 method comprising:
4 dividing the virtual machine instruction sequence into basic blocks each
5 corresponding to an instruction block;
6 transmitting the instruction block to the virtual machine;
7 a storing step of storing the instruction block in the virtual machine;

8 formatting the instruction block to include:

9 an identifier area for storing an identifier that specifies a start position of
10 the instruction block;

11 a non-branch instruction area for storing non-branch instructions
12 belonging to the corresponding basic block;

13 a branch instruction area for storing at least one branch instruction
14 belonging to the corresponding basic block; and

15 each branch instruction stored in the branch instruction area designating a
16 branch destination using an identifier stored in one of the identifier areas,

17 wherein the division of the virtual machine instruction sequence into a
18 plurality of separately identifiable instruction blocks having a single branch
19 instruction area reduces the amount of branch destination processing that
20 would otherwise be necessary with a single instruction sequence with branch
21 instructions throughout.

1 53. (Previously Presented) A method of storing a virtual machine
2 instruction sequence generated by compiler to be executed by a virtual machine, the
3 improvement method comprising:

4 dividing the virtual machine instruction sequence into basic blocks each
5 corresponding to an instruction block;

6 transmitting the instruction block to the virtual machine;

7 storing the instruction block in the virtual machine;

8 formatting the instruction block to include:

9 an identifier area for storing an identifier that specifies a start position of
10 the instruction block;

11 a non-branch instruction area for storing non-branch instructions
12 belonging to the corresponding basic block;
13 a branch instruction area for storing at least one branch instruction
14 belonging to the corresponding basic block; and
15 each branch instruction stored in the branch instruction area designating a
16 branch destination using an identifier stored in one of the identifier areas,
17 wherein the virtual machine instruction sequence is transmitted after being
18 divided into the plurality of instruction blocks.